

Introduction

In this module, we will cover selection statements, or those statements in Python which help make choices during the program run-time and select a set of actions based on a specific criterion or based on the flow of data.

We have two topics to cover in this module: the first is **comparison expressions**, also known as **relational and logical expressions**, statements in Python that compare two operands or variables, and result in either a Boolean or NoneType object. Then we will move to the second topic in which we will learn the different **types of selection statements**. For each type, the syntax and structure will be covered, and an illustrative example will be provided to help in exploring them in practice.

Topics and Learning Objectives

Topics

- Comparison expressions
- Types of selection statements

Learning Objectives

By the end of the module, you will be able to:

- Relate the third-valued logic to the output of the conditional statements.
- Construct a conditional statement to make decisions at run-time.
- Differentiate between one-way, two-way, and multi-way statements.

Readings

Required Readings

- Lambert, K. A. (2019). *Fundamentals of Python: First programs* (2nd ed.). Boston, MA: Cengage Learning.
 - Loops and Selection Statements, pages 77–85

Comparison Expressions

Please watch and study this video presentation as it demonstrates the implementation of the relational and logical operators in Python. It also explains the critical differences between the relational and logical expressions in terms of designated operators and the data type of their respective operands. The standard Boolean logic used in Python, which includes the **True** and **False** objects, has been extended to include a third logical object: **None**, which represents the absence of data and interprets the output of the logical expressions.

Ryecast Video:

CIND830 - Module 4: Comparison Expressions

Download

Download the PDF file for [CIND830 - Module 4: Selection Statements](#).

Please note this PDF file will be used in the two videos of this module. In the PDF, you can use the top navigation bar to navigate between the topics covered in each video.

Download the [CIND830 - Module 4: Comparison Expressions transcript](#).

Types of Selection Statements

Please watch and study this video presentation. It explains how the Python interpreter selects specific statements or actions, and makes choices during run-time, or as data flow in the program. Three types of selection statements will be explored: **one-way**, **two-way**, and **multi-way** selection statements. For each type, an illustrative example is provided to show the selection statement's syntactic structure and describe the boundaries of the statement at run-time.

Ryecast Video:

CIND830 - Module 4: Types of Selection Statements

Download

Download the PDF file for [CIND830 - Module 4: Selection Statements](#) and click on the topic "**Types of Selection Statements**" in the top navigation bar to take you to the beginning of this topic.

Download the [CIND830 - Module 4: Types of Selection Statements transcript](#).

Self-Assessment Quiz

Please complete the following review activity. The activity is not graded.

Test Yourself

H5P Object:

Self-Assessment Quiz

Summary

In this module, we covered the comparison expressions that include relational and logical expressions. We highlighted the key differences between them and introduced the three-valued logic that Python uses as an output for the logical expressions. It is worth knowing how users encode missing data in their program to ensure accuracy in results. Then, we moved to explore the syntactic structure of different types of selection statements in Python. We were keen to demonstrate our examples in both control and data flow formats to ensure clarity and understanding of how the Python interpreter generates results.

Activities

Discussion

Module 4 Discussion

This week, let's discuss conditional statements with one-way, two-way, and multi-way selection statements in the **Module 4 Discussion** topic in D2L under the **Communication** drop-down menu, then **Discussions**. Feel free to start a new thread or respond to someone else's post.

Lab Activity

Lab 3

Before starting Lab 3, be sure you have reviewed all the content in Module 4. The lab is a practical extension to apply your knowledge gained from this module.

[Download Lab 3](#) now and complete it by the end of this week. Your instructor will give you time to work on it before releasing the solutions in a folder called **Lab Solutions** in D2L under the **Course Materials** drop-down menu for you to verify your answers. The lab activity is not graded.

While working on your lab, post any questions, ideas, and strategies in the **Module 4 Discussion** topic in D2L. Feel free to read and respond to your peers' posts as well. You may also have the option of reaching out to your instructor during his/her virtual office hours.

Please note: The PDF for the lab activity may not be fully accessible for students who use assistive technology. If you require an accessible version, please contact your instructor.

Assessments

Quiz Module 4

Please go to your D2L course shell, under the **Assessment** drop-down menu, and select **Quizzes**. Then, click on the **Quiz Module 4** link and follow the instructions to complete this quiz. The date and length of time have been set for this quiz. **This quiz is a cumulative 10% of your final grade.**

Reminder

Assignment 1 (worth 15% of your final grade) is due by the end of Module 5. Remember to submit both the notebook file (in IPYNB format) and the exported version (either in PDF or HTML format) in the same **Assignment 1** link in D2L. Failing to submit both files will be subject to a mark deduction.

While you are working on this assignment, feel free to share your questions, ideas, and strategies in the **Assignment 1 Discussion** topic in D2L.

References

- Lambert, K. A. (2019). *Fundamentals of Python: First programs* (2nd ed.). Boston, MA: Cengage Learning.
- Python Software Foundation. (2020). [Python 3.7.8 documentation](https://docs.python.org/3.7/index.html). Retrieved from <https://docs.python.org/3.7/index.html>.